



TITLE:

多値論理式の簡単化アルゴリズム について (多値論理およびその応用)

AUTHOR(S):

笹尾, 勤; 石川, 啓二; 寺田, 浩詔

CITATION:

笹尾, 勤...[et al]. 多値論理式の簡単化アルゴリズムについて (多値論理
およびその応用). 数理解析研究所講究録 1982, 455: 64-75

ISSUE DATE:

1982-03

URL:

<http://hdl.handle.net/2433/103025>

RIGHT:

多値論理式の簡単化アルゴリズムについて

阪大 工学部 笹尾 勤

石川啓二

寺田若詔

1. まえがき

LSIの集積度が上昇するにつれて、設計工数が急激に増大するようになった。そのため、従来、人手によって行われてきた論理設計も、設計期間を減らし、設計誤りを無くするために、自動化が必須になってきている。

プログラマブル・ロジック・アレイ(PLA)は構造が規則的であるため、自動設計の研究が最も進んでおり、今までに多くのPLAが、LSIやVLSIの一部で使用されている[1]~[3]。

従来の2段PLAは、AND-OR 2段多出力論理回路に対応し、その最小化は、二値多出力関数を表現する論理式の最小化問題に帰着できる。

2段PLAにデコーダを付加した、デコーダ付きPLAは、同じ関数を、2段PLAよりも10~20%小さなアレイで実

現でできることから[4]，既に多数のものが使用されており，使用者が自由に書き込めるFPLAも市販されている[1]～[3]。

デコーダ付きPLAの最小化問題は，多値入力二値関数を表現する式の最小化問題に帰着できる[4]～[6]。この問題は，基本的には，二値論理式の最小化と同様な方法，すなわち，主項を生成し，最小被覆を求めるという古典的な方法（クワイン－マクラスキの方法）で解決できる。

LSIやVLSIで使用するPLAは，入力数や出力数が数十のものが多く，能率の良い論理式簡単化アルゴリズムが望まれている。

同じ二値論理関数を表現する，二値論理式と多値入力二値関数の式を比べると，多値の式の方が，主項の個数が多く，かつ必須主項の個数が少ないので，最小化に要する手数は多い[4]。

今までに発表されている論理式簡単化アルゴリズムは，その手法から次のように分類できる。

方法1： すべての主項を同時に生成[7]～[10]。

方法2： キューブの周辺のすべての主項を生成[11]。

方法3： 節点を被覆するすべての主項を生成[12]～[14]。

方法4： 各節点に関する情報を求める[15]～[17]。

方法5: 繰り返し改善法 [18]。

方法6: キューブ変形法 [9], [19]。

方法7: その他 [20]。

このうち、方法1は、主項の個数が、二値の場合最悪 3^n のオーダで、多値の場合さらに急激に増大することにより [4], n が大きいとき使用不能である。同様な理由で、方法2及び方法3も実用的ではない。方法4は、 $f = 1$ となる入力の各組合せに対して、“組合せの良さ”を記憶する方法であり、一般には、 2^n のオーダのメモリ容量が必要となる。 $(n = 16$ までの実用プログラムはあるが [15], それより大きい問題は取り扱えない。)

方法5は、PLA設計用に、IBMで開発されたものであり、多値論理式の簡単化も可能であり、いくつかのメーカーで使用されている。方法6は、 n が大きくなっても、使用メモリや計算時間が急激には増大しないので、大規模なPLAの設計に向いており、一部メーカーで使用されている。方法7は、非常に高速に非冗長解を求めることができるが、あまり良い解は期待できない。

このように、論理式簡単化アルゴリズムは種々のものが公表されているが、いくつかの実用アルゴリズムを比較したものは、ほとんどない [9]。本論文では、8個のアルゴリズムの性能を、計算時間と解の良さの点から比較する。

2. 定義および基本的性質

定義 2.1 $P_i = \{0, 1, \dots, p_i - 1\}$, $B = \{0, 1\}$ とするとき,
 $f: \prod_{i=1}^n P_i \rightarrow B$ を, 多値入力二値関数 という。

補題 2.1 多値入力二値関数は次式 (以下では多値論理式という) で表現できる。 $S_i \subseteq P_i$ とするとき,

$$f(X_1, X_2, \dots, X_n) = \bigvee_{(S_1, S_2, \dots, S_n)} X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}.$$

$$\text{ここで, } X_i^{S_i} = \begin{cases} 1 & (X_i \in S_i) \\ 0 & (X_i \notin S_i) \end{cases} \quad \text{である。}$$

定義 2.2 $u = |f^{-1}(1)|$ を f の重み, $d = u / (\prod_{i=1}^n p_i)$ を 真理値表濃度 という。

定義 2.3 $X_1^{S_1} X_2^{S_2} \dots X_n^{S_n} \leq f$ のとき, $X_1^{S_1} X_2^{S_2} \dots X_n^{S_n}$ を, f の項という。 f の項のうち, $|S_i| = 1$ ($i = 1, 2, \dots, n$) であるものを f の最小項, 極大な項を f の主項という。 f を表現する式のうち, 項数極小のものを非冗長形式, 項数最小のものを最簡形式という。主項のみから成る最簡形式を主最簡形式という。すべての主最簡形式に含まれる主項を必須項という。

3. 簡単化アルゴリズム

多値論理式の簡単化は二値の場合と同様に行なえるが, ニ

値の場合に比べ、主項の個数が多く、必須項の個数が少ない。

1) 古典的方法 (Quine - Mc Cluskey 法)

QM i) f の主項をすべて生成する (Tison の方法)。

ii) 主項表を作る。

iii) 主項表の最小被覆を求める (分枝限定法)。

最小被覆を最後まで実行するプログラムを QM1, 最初の解が求まった時点で停止するものを QM2 で表わす。

2) 繰り返し改善法

MINI f と \bar{f} の分離的論理和形 (disjoint disjunctive form) を求め, f の各項に対し次の三つの操作を繰り返す。

i) f を変化させずに各項を縮小する。

ii) 二つの項に対し, 一方を拡大, 他方を縮小する。

iii) 各項を主項まで拡大する。他の項に被覆される項は除去する。

3) キューブ変形法

A1 i) 各項を主項まで拡大する。

ii) 冗長な項を除去する。

A2 i) f の各項に対し, 次の変形を行なう。

a) 項をできるだけ縮小する

b) 未被覆の最小項をできるだけ多く含む方向に

拡大する。

ii) 冗長な項を除去する。

- A3 i) (擬似) 必須項を求め, その項を解に含める。
 ii) (擬似) 必須項と共通部分をもつか, 隣接する項を求め, 変形する。
 iii) 上記操作を繰り返す。(擬似) 必須項が求まらないときには, 未被覆の最小項をできるだけ多く含む項を必須項と考える。
 iv) 冗長な項を除去する。

- A4 i) 各項を主項まで拡大する。
 ii) 冗長な項を除去し, F の項を二つに分類する。
 F_1 : 必須項になる可能性のある項
 F_2 : 非必須項
 iii) F_1 の項を変形し, F_2 の項を変形する。
 iv) 冗長な項を除去する。

- A5 i) 各項を主項まで拡大する。
 ii) F の項を四つに分類する。
 F_1 : 必須項
 F_2 : F_1 と共通部分をもつ項
 F_3 : F_1 と隣接する項
 F_4 : その他

- iii) F_1 を解に加える。 $F_2 = F_3 = F_4 = \emptyset$ ならば, ivへ。
 $F_2 = F_3 = \emptyset$ ならば, F_4 の一つを必須項として,
 iiへ戻る。
 F_2, F_3 の項を順に変形する。変形した項を必須項とし, iiに戻る。
 iv) 冗長な項を除去する。

4. 実験結果

前述のアルゴリズムをFORTRANで記述し, ACOS 900 (大阪大学大型計算機センタ) を用いて, 種々の問題を簡単化した結果を, 表4.1~表4.4に示す。表中, 最初の数字は積項数, 次の数字は計算時間(秒)を示す。

- 1) QM1とQM2は, 真理値表濃度 d が小さいときには, 最小解が短時間に求まる。しかし, d が大きくなると, 計算時間が急速に増加する(表4.1)。QMは, $\sum_{i=1}^n \log_2 p_i \leq 10$ ならば, (擬似)最小解が比較的短時間に求まると推定できる。
- 2) MINIは, すべての種類の問題で良い解が求まった。
 d が小さい場合は他のアルゴリズムよりも時間がかかるが, d が大きい場合には短時間である(表4.1)。表4.4の

M-156の問題に対しては、メモリオーバーとなり（配列領域が130k語を越えた）、実行不能であった。

- 3) A1は他のアルゴリズムよりも高速であるが、解は最も悪い。特に多出力の問題に対しては、最小解から大きくずれている。
- 4) A2の解はMINIより悪く、計算時間は中程度である。
- 5) A3は、小さい問題に対しては、QM1, QM2と同程度に良い解を生じた。dが大きくなるにつれて、計算時間が急激に増大した（表4.1）。大きな実用回路に対しては、計算時間オーバーとなることが多かった。
- 6) A4はA2よりも良い解を生ずるが、計算時間が長くなることが多い。
- 7) A5はA4より高速であり、解の最小性を証明できた場合もあった（表4.4の問題M-156）。

表4.1 乱数により生成した10変数論理関数に対する
各アルゴリズムの比較

d	1/8	2/8	3/8	4/8	5/8	6/8	7/8
IN-OUT	10-1	10-1	10-1	10-1	10-1	10-1	10-1
F-DC	128-0	256-0	384-0	512-0	640-0	768-0	896-0
2-valued	QM1	#91- 2.5	#131-12.9	#156-45.9	*166-289	*155-733	*141-2192
	QM2	#91- 2.5	#131-12.9	157-45.7	167-159	157-400	142-1182
	MINI	91-52.9	134-131	164-190	175-211	165-273	151- 222
	A1	91- 9.5	139-28.6	175-54.3	189-80.3	184-97.5	177- 119
	A2	91-13.6	134-46.6	164-89.5	178-138	177-170	164- 191
	A3	#91-12.1	131-58.0	161-210	+171-407	+175-498	+166- 602
	A4	91-16.4	136-55.5	167-103	174-155	170-171	157- 192
4-valued	A5	91-14.0	134-48.0	165-89.2	180-124	175-154	164- 175
	QM1	#75- 1.6	#110-10.5	*121-114	*122-595	*116-2443	
	QM2	#75- 1.6	#110-10.5	123-64.1	122-328	118-1304	
	MINI	77-19.9	111-53.8	126-86.7	132-120	126-116	109-117
	A1	77- 4.9	120-15.3	143-26.1	159-38.2	155-47.3	129-51.7
	A2	76- 6.6	114-20.9	130-36.1	133-54.6	132-76.7	118-99.7
	A3	75- 7.7	110-44.5	+126-133	+147-199	+148-245	+128-286
	A4	77- 8.1	115-28.6	129-44.8	132-68.1	128-89.0	114-97.7
	A5	75- 7.7	115-26.4	130-44.5	144-68.0	133-94.3	115-103
							71-77.3
							84-52.2
							68-102
							+84-385
							77-100
							72-88.4

* denotes minimum covering has not been completed.

† A3 was stopped in about 5 times as the time spent for A1.

denotes the solution was proved to be minimum.

表4.2 演算回路に対する各アルゴリズムの比較

	$f=5x+1$ (mod 256)	$f=\sqrt{x}$	$f=\sqrt{x^2+y^2}$	$f=x+y$	$f=xy$	$f=x^2$	$f=\lfloor \frac{255}{\log(x+1)} \rfloor$ /log256]
IN-OUT	8-8	8-5	8-5	8-5	8-8	6-12	8-8
F-DC	255-0	255-0	255-0	255-0	225-0	63-0	255-0
2-valued	QM1	#76-386	#57-89.7	#120-143	#75-191	*123-541	*47-47.8
	QM2	#76-386	57-89.7	#120-143	#75-191	123-279	48-25.7
	MINI	76-75.4	57-28.3	121-112	75-58.8	129-177	50-25.0
	A1	135-85.1	58-12.0	129-36.1	91-79.8	152-60.6	62- 4.9
	A2	77-103	58-21.6	122-65.7	75-67.7	134-94.3	54-14.6
	A3	+80-287	57-74.9	121-242	75-176	+135-292	52-85.4
	A4	77-94.3	57-34.2	121-103	75-79.6	131-146	50-21.8
4-valued	A5	76-75.7	58-29.4	121-84.1	75-70.5	134-126	51-20.8
	QM1	#48-809	#40-122	*94-495	#17-31.0	*88-349	#39-62.1
	QM2	#48-809	#40-122	97-256	#17-31.0	88-241	40-37.1
	MINI	53-32.8	42-12.9	101-62.8	17- 8.2	94-63.8	46-11.9
	A1	93-68.7	44- 8.1	115-22.7	18-12.9	122-34.8	58- 4.2
	A2	50-115	42-16.4	99-43.2	17-13.0	92-61.3	45-13.0
	A3	+52-297	41-57.3	+101-283	17-14.7	+90-282	47-85.2
	A4	50-52.1	41-25.4	100-68.6	17-10.2	92-72.6	44-19.2
	A5	49-46.3	41-19.9	101-57.5	17- 9.8	91-63.4	44-16.3
							114-65.0
							130-28.9
							112-136
							115-206
							115-135

* denotes minimum covering has not been completed.

† A3 was stopped in 280 seconds (CPU time limit).

denotes the solution was proved to be minimum.

表4.3 実用回路に対する各アルゴリズムの比較

		7-segment display	Add- subtract	Control circuit	Micro- program	Black Jack	Hollerith ASCII	Symmetric function
	IN-OUT	10-23	18-15	18-26	16-8	10-14	14-8	6-1
	F-DC	40-0	32-0	34-0	147-0	40-2	18-0	50-0
2-valued	QM1							#15-2.0
	QM2							#15-2.0
	MINI	35- 9.7	32-11.4	34-18.2	55-54.2	18-3.6	18-12.0	15-1.6
	A1	36-12.3	32- 9.8	34- 3.2	62-46.5	19-1.0	18- 0.7	21-1.0
	A2	36-30.2	32-15.4	34- 4.8	57-54.2	19-2.2	18- 0.9	18-2.8
	A3	---	---	#33- 9.4	+60-288	#18-3.1	#18- 0.8	16-7.9
	A4	35-48.1	32-32.8	33-11.3	55-62.1	18-5.8	18- 1.7	18-3.2
4-valued	A5	36-33.5	32-23.7	33- 8.2	56-56.2	18-4.4	#18- 1.6	16-2.4
	QM1							#6-0.3
	QM2							#6-0.3
	MINI	35-29.2	29- 6.0	33-10.6	45-32.6	17-1.8	18- 3.4	6-0.3
	A1	34- 7.8	29- 5.8	34- 2.3	53-33.1	18-0.8	18- 0.5	8-0.2
	A2	33-21.3	29-11.8	34- 4.3	50-48.4	18-3.4	18- 0.7	6-0.5
	A3	---	---	#33- 8.5	+46-277	18-3.9	#18- 0.6	6-0.9
	A4	33-32.0	29-23.7	33-10.9	47-45.2	18-9.7	18- 1.5	8-0.8
	A5	33-19.5	29-16.3	33- 7.1	48-37.4	18-5.1	18- 1.4	6-0.5

† A3 was stopped in 280 seconds (CPU time limit).

denotes the solution was proved to be minimum.

--- denotes excessive computation time.

表4.4 実用回路に対する各アルゴリズムの比較

		ROM M-1024	Control circuit M-001	Control circuit M-004	ROM M-110	Clock M-1010	Clock M-1011	Micro program M-672	Control circuit M-156
	IN-OUT	10-4	14-23	16-14	10-4	12-7	12-8	17-13	34-4
	F-DC	1024-0	71-0	59-0	511-0	77-0	132-0	672-0	180-0
2-valued	QM1	-----							
	QM2	-----							
	MINI	16-164	54-44.3	31-25.7	58-56.4	47-28.7	112-115	182-1642	
	A1	16-165	53-41.6	35-18.0	58-39.4	47- 9.9	117-23.7	220- 712	112-1309
	A2	16-189	55-181	31-39.3	58-53.4	48-24.3	113-50.6	185- 953	
	A3	#12-185		---	+58-276	45-144	112-145		
	A4	12-169	53-313	35-76.2	58-68.8	46-49.8	112-109	183- 918	
4-valued	A5	12-176	53-141	35-49.0	58-64.2	46-38.5	112-76.0	184- 781	
	QM1	-----							
	QM2	-----							
	MINI	11-80.6	51-28.0	28- 9.1	38-30.1	40-14.8	103-72.6	163- 683	-----
	A1	16-79.3	52-28.1	31- 8.9	40-24.9	42- 9.2	104-15.3	202- 441	97- 737
	A2	11-90.1	52-181	30-32.1	38-34.2	40-41.9	102-36.9	167- 797	97-3222
	A3	#11-86.3		---	38-46.0	+41-316	101-89.6		
	A4	11-86.1	52-340	30-62.1	38-39.2	41-74.2	102-78.1	162- 646	
	A5	11-83.3	52-135	30-34.3	38-36.4	39-45.7	102-58.6	163- 540	#97-2283

† A3 was stopped in 280 seconds (CPU time limit).

denotes the solution was proved to be minimum.

--- denotes excessive computation time.

----- denotes excessive memory space.

[文 献]

- [1] H.Fleisher and L.I.Maissel, "An introduction to array logic," IBM J. Res. Develop., vol.19, pp.98-109, March 1975.
- [2] P.W.Cook, C.W.Ho, and S.E.Schuster, "A study in the use of PLA-based macros," IEEE J. Solid-State Circuits, vol.SC-14, No.5, pp.833-838, Oct. 1979.
- [3] R.L.Golden, P.A.Latus, and P.Lowy, "Design automation and the programmable logic array macro," IBM J. Res. Develop. vol.24, No.1, pp.23-31, Jan. 1980.
- [4] T.Sasao and H. Terada, "Multiple-valued logic and the design of programmable logic arrays with decoders," Proceedings of ISMVL-79, pp.27-37, May 1979.
- [5] T. Sasao, "An application of multiple-valued logic to a synthesis of programmable logic arrays," Proceedings of ISMVL-78, pp.65-72, May 1978.
- [6] T. Sasao and H. Terada, "On the complexity of shallow logic functions and the estimation of programmable logic array size," Proceedings of ISMVL-80, pp.65-73, June 1980.
- [7] E.J.McCluskey, Jr. "Minimization of Boolean functions," Bell System Tech. J. 35, 1417, 1965.
- [8] E.J.McCluskey, Jr. Introduction to the Theory of Switching Circuits, McGraw-Hill, 1965.
- [9] M.A.Breuer (Ed.), Design Automation of Digital Systems, vol.1 : Theory and Techniques, Prentice-Hall, 1972.
- [10] M.Auguin, F.Boeri, and C.Andre, "An algorithm for designing multiple Boolean functions: Application to PLA's.," Digital Process, N.4, pp.215-230, 1978.

- [11] J.P.Roth, "Algebraic topological methods in synthesis," in Proceeding of International Symposium on Theory of Switching, April 1957, in Annals of Computational Laboratory of Harvard University, vol.29, pp.57-53, 1959.
- [12] V.T.Rhyne, P.S.Noel, M.H.McKinney, and U.W.Pooh, "A new technique for the fast minimization of switching function, " IEEE Trans. Comput. vol. C-26, No.8, pp.757-764, August 1977.
- [13] R.S.Michalski and Z.Kulpa, "A system of programs for the synthesis of combinatorial switching circuits using the method of disjoint stars," Proceedings of International Federation of Information Processing Societies Congress, 1971, Booklet TA-2, p.158, 1971.
- [14] H.T.Nagale Jr., B.D.Carroll, and J.D.Irwin, An Introduction to Computer Logic, p.168, Prentice-Hall, Inc. ,1975.
- [15] P.Bricaud and J.Campbell, "Multiple output PLA minimization: EMIN," WESCON 78, 33/3, 1978.
- [16] Z.Arevalo and J.G.Bredeson, "A method to simplify a Boolean function into a near minimal sum-of-products for programmable logic arrays," IEEE Trans. on Comput. Vol.C-27, No.11, pp.1028-1039, Nov. 1978.
- [17] A.K.Halder, "Grouping table for the minimization of n-variable Boolean functions," Proc. IEE, vol.125, No.6, pp.474-482, June-1978.
- [18] S.J.Hong, R.G.Cain, and D.L.Ostapko, "MINI: A heuristic approach for logic minimization," IBM J. Res. Develop. vol.18, pp.443-458, Sept. 1974.
- [19] D.W.Brown, "A state-machine synthesizer-SMS", Proceedings of 18th Design Automation Conference, pp.301-305, July 1981.
- [20] G.Pomper and R.J.Armstrong, "An efficient multivalued minimization algorithm," Proceedings ISMVL-79, May 1979.